

# Base de Datos Distribuidas

*Mamani Aliaga, Jaime Ysaac*

ysaacx@episunsa.edu.pe

CUI:20031070

*Poco Medina, Jorge Luis*

jpocom@episunsa.edu.pe

CUI: 20031077

26 de octubre de 2006

## 1. Escribir el procedimiento para utilizar las características de Base de Datos Distribuidas en *Oracle* y *Postgres*

### 1.1. Oracle

En la arquitectura cliente-servidor, el sistema de base de datos Oracle se divide en 2 partes: (1) la maquina de la parte visible al usuario (maquina front-end) como parte cliente, y (2) la maquina del sistema subyacente (maquina back-end) como parte servidor. La parte cliente es la aplicación de base de datos que interactúa con el usuario. El cliente no tiene la responsabilidad de acceso a datos y maneja meramente la petición, procesamiento y presentación de los datos gestionados por el servidor. la parte servidor ejecuta Oracle y maneja las funciones relativas al acceso compartido concurrente. Acepta sentencias SQL y PL/SQL originadas por aplicaciones del cliente, las procesa, y devuelve los resultados al cliente. las aplicaciones cliente-servidor Oracle proporcionan transparencia de localización haciendo la localización de los datos transparentes al usuario; algunas características como vistas, sinónimos y procedimientos contribuyen a proporcionar esa transparencia. Se consiguen nombres utilizando  $\langle \text{nombre tabla} . @ , \text{nombre base datos} \rangle$  para referirse de manera única a las tablas.

Oracle utiliza un protocolo de confirmación en dos fases para gestionar las transacciones distribuidas concurrentes. La sentencia COMMIT dispara el mecanismo de confirmación en dos fases. El proceso *background* RECO (recuperar) resuelve automáticamente el resultado de estas transacciones distribuidas en las que se interrumpe el COMMIT. El proceso RECO de cada Servidor Oracle local confirma o deshace automáticamente cualquier transacción distribuida  $\ll \text{enduda} \gg$  consistentemente en todos los nodos involucrados. Para fallos de larga duración, Oracle permite a cada BD local confirmar o deshacer manualmente cualquier transacción en duda y liberar los recursos. La consistencia global se puede mantener restaurando la base de datos en cada sitio retornando a un punto fijo predeterminado del pasado.

La arquitectura de la base de datos distribuidas de Oracle se muestran en la Figura 1 Un nodo de un sistema de base de datos distribuido puede actuar como un cliente, como un servidor, o como ambos, dependiendo de la situación. La figura muestra dos sitios donde están las bases de datos llamadas OC (oficina central) y Ventas. Por ejemplo, en la aplicación mostrada que se ejecuta en la oficina central, para una sentencia SQL emitida contra datos locales (por ejemplo, DELETE FROM DEPT ...), el computador OC actúa como un servidor, mientras que para una sentencia contra datos remotos (por ejemplo, INSERT INTO EMP@VENTAS), el computador OC actúa como cliente.

Todas las bases de datos Oracle en un sistema de base de datos distribuidos (SBDD) utiliza el software de red Oracle Net8 para comunicacion entre bases de datos. Net8 permite a las bases de datos comunicarse a través de redes para soportar transacciones distribuidas y remotas. Empaqueta sentencias SQL en uno de los muchos protocolos de

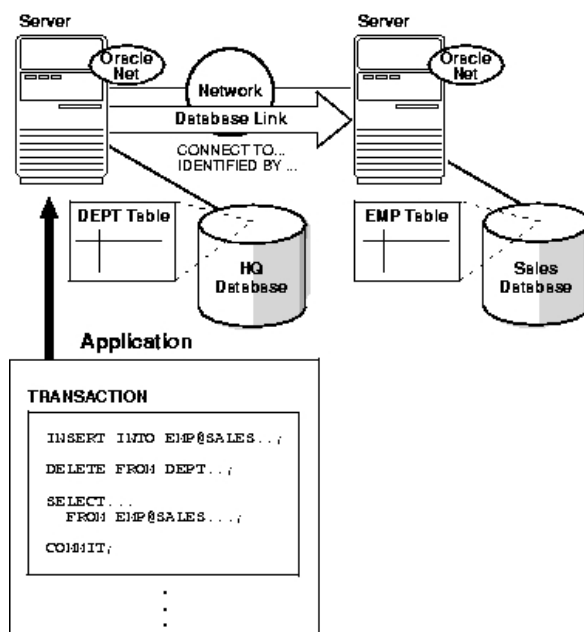


Figura 1: Sistema de base de datos distribuidas en Oracle

comunicacion para facilitar al cliente la comunicacion con el servidor y despues empaqueta y devuelve los resultados de forma similar al cliente. Cada base de datos tiene un nombre global unico proporcionado por una ordenacion jerarquica de nombres de dominio de red que se prefija antes de dar nombre la base de datos para hacerlo unico.

Oracle soporta enlaces de base de datos que definen un camino de comunicacion de un sentido, entre dos bases de datos Oracle. Por ejemplo,

```
CREATE DATABASE LINK ventas.us.americas,
```

establecer una coneccion a la base de datos ventas de la figura 1 bajo el dominio de red *us* que viene bajo el dominio *americas*.

Los datos de un SBDD Oracle se pueden replicar utilizando instantaneas (fotograficas) o tablas principales replicadas. La replicacion se proporciona en los siguientes niveles:

- *Replicación básica:* las replicas de tablas se gestionan para accesos de solo lectura. para modificaciones, se deberá acceder a los datos del sitio primario.
- *Replicación avanzada (simetrica):* ésta es una extension sobre la replicacion basica, que permite a las aplicaciones que modifiquen replicas de tablas por todo un SBDD replicado. Los datos se pueden leer y modificar en cualquier sitio. Esto requiere un software adicional llamado opcion de replicacion avanzada de Oracle. Una instantanea genera una copia de parte de la tabla por medio de una consulta llama *consulta de definicion de la instantanea*. Una definicion simple de una instantanea seria la siguiente:

```
CREATE SNAPSHOT ventas.pedidos AS
SELECT * FROM ventas.pedido@oc.us.americas;
```

## 1.2. Postgres

Postgres como sistema de Base de datos tiene una excelente carta de presentación con respecto a otras, tiene un mercado amplio, ya que es de distribución libre (código abierto).

Postgres ha innovado su tecnología constantemente, al punto de estar intentando creando bases de datos distribuidas, para lo cual la información con respecto al desarrollo de dicha tecnología aun es pobre, pero con mucha tendencia a el crecimiento.

Podemos indicar a continuación ciertas forma de procesamiento distribuido.

```
SELECT * FROM dblink(Œdbname=mydbŒ, Œselect pronom, prosrc from pg_procŒ)  
AS t1(pronom name, prosrc text)
```

La sentencia anterior logra ejecutar consultas remotas, a otras bases de datos

Tenemos que aclarar que existen herramientas para replicar datos en otras maquina, pero no podemos decir que es un sistema de base de datos distribuido puro, ya que aunque cuenta con ciertos criterios no lo es completamente.

También podemos decir que que **dblink** sirve como una forma de ordenar la información, algo así como se utilizan los esquemas de el postgres.

### 1.2.1. Ejemplos

Algunos Ejemplos de uso de **dblink**:

Hacer consultas con **Select**:

```
select * from dblink  
(  
    'dbname=pgteste  
    hostaddr=192.168.1.1  
    user=postgres  
    password=postgres  
    port=5432',  
    'select nome  
    from clientes  
    ',  
    ) as t1(nome varchar(30));
```

Hacer consultas con **Insert**:

```
select  
    dblink_exec(  
        'dbname=pgteste  
        hostaddr=192.168.1.1  
        user=postgres  
        password=postgres  
        port=5432',  
        'insert into clientes(nome)  
        values(''roger'')  
        ',  
    );
```